



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/520,949	12/06/2005	Mark Frazer	213222.00096	4906
27160 7590 03/16/2010 KATTEN MUCHIN ROSENMAN LLP (C/O PATENT ADMINISTRATOR) 2900 K STREET NW, SUITE 200 WASHINGTON, DC 20007-5118			EXAMINER KIM, TAE K	
			ART UNIT 2453	PAPER NUMBER
			MAIL DATE 03/16/2010	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/520,949

Applicant(s)

FRAZER, MARK

Examiner

TAE K. KIM

Art Unit

2453

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 January 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 12-38 and 70 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 12-38 and 70 is/are rejected.
- 7) ☒ Claim(s) 26, 27, 36, and 37 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB06)
Paper No(s)/Mail Date 05/04/08
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

This is in response to the application filed on December 5, 2005 where Claims 1 – 11 and 39 – 69 have been cancelled. Claims 12 – 38 and 70, of which Claims 12, 29, and 70 are in independent form, are presented for examination.

Priority

Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on May 4, 2008 was filed after the mailing date of the U.S. Application on January 12, 2005. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Objections

1. Claims 26, 27, 36, and 37 are objected to because of the following informalities: does not end with a period. Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 12 – 38 and 70 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 7,143,131, invented by L. Roger Soles, et al. (hereinafter “Soles”).

2. Regarding Claim 12, Soles discloses of a client host computer for use in reliably transmitting a request wrapped in a request message over a computer network to a server protocol stack in a server host computer [Fig. 1; Col. 4, lines 38-60], the client host computer having a client protocol stack for receiving the request from a client application [Fig. 1, item 112; Col. 4, lines 38-60], wrapping the request into a request message, and transmitting the request message to the server host computer [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], the client protocol stack when so doing calling a set of functions external to the client protocol stack and that are provided by the client application [Col. 9, lines 47-67], the set of functions providing reliable transport-related services to the client protocol stack that may pre-selected to meet the requirements of the client application [Col. 9, lines 47-67].
3. Regarding Claim 13, Soles discloses all the limitations of Claim 12 above. Soles further discloses that the set of functions provided to the client protocol stack includes a function for determining a retransmission timer interval for an i-th retransmission of the request message and a total allowed number of transmissions of the request message to be made before an error is reported [Col. 9, lines 47-67].
4. Regarding Claim 14, Soles discloses all the limitations of Claim 13 above. Soles further discloses that the set of functions provided to the client protocol stack includes a function that the client protocol stack may call to wrap the request in the request

message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

5. Regarding Claim 15, Soles discloses all the limitations of Claim 14 above. Soles further discloses that the function that the client protocol stack may call to wrap the request in the request message adds a header to the request message including an identifying sequence number assigned by the client protocol stack and a type code identifying the type of the request message as a request [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

6. Regarding Claim 16, Soles discloses all the limitations of Claim 15 above. Soles further discloses that the set of functions provided to the client protocol stack includes a function that the client protocol stack may call to obtain a starting sequence number and range of allowable sequence numbers for request messages [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

7. Regarding Claim 17, Soles discloses all the limitations of Claim 16 above. Soles further discloses that the set of functions provided to the client protocol stack includes a function that the client protocol stack may call to parse a message received from the server protocol stack and obtain the type and sequence number of that message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

8. Regarding Claim 18, Soles discloses all the limitations of Claim 17 above. Soles further discloses that the client protocol stack retransmits the request message if a reply message is received from the server protocol stack that contains the sequence number of the request message is not received before the retransmission timer interval for i-th transmission has elapsed since the request message was transmitted for the i-th time,

and the number of times that the request message has been transmitted is less than the total allowed number of transmissions, but if a provisional reply message is received from the server protocol stack that contains the sequence number of the request message, then expiration of the current retransmission timer interval is delayed [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

9. Regarding Claim 19, Soles discloses all the limitations of Claim 18 above. Soles further discloses that the server host computer has a server protocol stack for receiving a request message transmitted from the client host computer [Fig. 1, item 102], extracting a request wrapped in the request message, providing the request to a server application running on the server host computer, wrapping a reply received from the server application into a reply message, and transmitting the reply message back to the client host computer, the server protocol stack when so doing calling a set of functions external to the server protocol stack and provided by the server application, the set of functions providing reliable transport-related services to the server protocol stack than may be pre-selected to meet the requirements of the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

10. Regarding Claim 20, Soles discloses all the limitations of Claim 19 above. Soles further discloses that the set of functions provided to the server protocol stack includes a function for determining a reply cache interval for a reply cache timer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

11. Regarding Claim 21, Soles discloses all the limitations of Claim 21 above. Soles further discloses that the set of functions provided to the server protocol stack includes

a function that the server protocol stack may call to wrap the reply into the reply message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

12. Regarding Claim 22, Soles discloses all the limitations of Claim 21 above. Soles further discloses that the function that the server protocol stack may call to wrap the reply in the reply message adds a header to the reply message including the identifying sequence number assigned by the client protocol stack to the corresponding request and a type code identifying the type of the reply message as a reply [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

13. Regarding Claim 23, Soles discloses all the limitations of Claim 22 above. Soles further discloses that if the message type of a message received from the client protocol stack indicates that the message contains a request, then the request is delivered to the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

14. Regarding Claim 24, Soles discloses all the limitations of Claim 23 above. Soles further discloses that if after delivery of the request to the server application the server application wishes to send a provisional reply, then a message is sent to the client protocol stack containing the sequence number and a message type code indicating that the message is a provisional reply to the request message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

15. Regarding Claim 25, Soles discloses all the limitations of Claim 24 above. Soles further discloses that when a reply message is sent to the client protocol stack, then the reply cache timer interval is started, the reply message is cached in a reply cache, and the cached reply message destroyed upon the expiration of the reply cache timer

interval [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

16. Regarding Claim 26, Soles discloses all the limitations of Claim 25 above. Soles further discloses that if the message type of a message received from the client protocol stack indicates that the message contains a request and a cached reply message has the same sequence number, then the cached reply message is resent to the client protocol stack [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

17. Regarding Claim 27, Soles discloses all the limitations of Claim 26 above. Soles further discloses that if the message type of a message received from the client protocol stack indicates that the message is an acknowledgement to a cached reply message, then only the sequence number in the cached reply message is retained in the reply cache [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

18. Regarding Claim 28, Soles discloses all the limitations of Claim 27 above. Soles further discloses that when the reply cache timer interval expires, then the cached reply message is deleted [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

19. Regarding Claim 29, Soles discloses a server host computer for use in providing reliable transport over a computer network [Fig. 1, item 102], the server host computer having a server protocol stack for receiving a request message transmitted from a client host computer, providing a request wrapped in the request message to a server application running on the server host computer, wrapping a reply received from the server application in a reply message, and transmitting the reply message back to the client host computer [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], the server protocol stack when so doing calling a set of functions external to the server protocol stack and

provided by the server application, the set of functions providing reliable transport-related services to the server protocol stack than may pre-selected to meet the requirements of the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

20. Regarding Claim 30, Soles discloses all the limitations of Claim 29 above. Soles further discloses that the set of functions includes a function for determining a reply cache interval for a reply cache timer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

21. Regarding Claim 31, Soles discloses all the limitations of Claim 30 above. Soles further discloses that the set of functions includes a function that the server protocol stack may call to wrap the reply into the reply message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

22. Regarding Claim 32, Soles discloses all the limitations of Claim 31 above. Soles further discloses that the function that the server protocol stack may call to wrap the reply in the reply message adds a header to the reply message including the identifying sequence number assigned by the client protocol stack to the corresponding request and a type code identifying the type of the reply message as a reply [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

23. Regarding Claim 33, Soles discloses all the limitations of Claim 32 above. Soles further discloses that if the message type of a message received from the client protocol stack indicates that the message contains a request, then the request is delivered to the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

24. Regarding Claim 34, Soles discloses all the limitations of Claim 33 above. Soles further discloses that if after delivery of the request to the server application the server

application wishes to send a provisional reply, then a message is sent to the client protocol stack containing the sequence number and a message type code indicating that the message is a provisional reply to the request message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

25. Regarding Claim 35, Soles discloses all the limitations of Claim 34 above. Soles further discloses that when a reply message is sent to the client protocol stack, then the reply cache interval is started, the reply message is cached in a reply cache, and the cached reply message destroyed upon the expiration of the reply cache interval [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

26. Regarding Claim 36, Soles discloses all the limitations of Claim 35 above. Soles further discloses that if the message type of a message received from the client protocol stack indicates that the message contains a request and a cached reply message has the same sequence number, then the cached reply message is resent to the client protocol stack [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

27. Regarding Claim 37, Soles discloses all the limitations of Claim 36 above. Soles further discloses that the message is an acknowledgement to a cached reply message, then only the sequence number in the cached reply message is retained in the reply cache [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

28. Regarding Claim 38, Soles discloses all the limitations of Claim 37 above. Soles further discloses that when the reply cache timer expires, then the cached reply message is deleted [Fig. 6g, 6h, item 670; Col. 11, lines 34-64].

29. Regarding Claim 70, Soles discloses a method for reliably transmitting over a

network a request from a client application running on a client system to a server application running on a server system [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], the method comprising, in the client system:

(a) when a command is received from the client application to open a client socket,

receiving from the client application a set of pointers to a set of personality functions provided by the client application, which include

an open function [Col. 4, line 37 – Col. 5, line 41],

a connect function [Col. 4, line 37 – Col. 5, line 41],

a wrap request function [Col. 4, line 37 – Col. 5, line 41],

a retransmit timer function [Col. 4, line 37 – Col. 5, line 41],

a parse function [Col. 4, line 37 – Col. 5, line 41],

a retransmit date update function [Col. 4, line 37 – Col. 5, line 41],

an acknowledgement reply function [Col. 4, line 37 – Col. 5, line 41], and

a close function [Col. 4, line 37 – Col. 5, line 41], and

calling the open function to obtain an initial sequence number, a range of allowable sequence numbers, a maximum number of retransmissions, and header and trailer sizes, and to allocate resources to needed to support the client socket [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(b) when a command is received from the client application to connect the client socket to the server application, calling the connect function to open a connection to the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(c) when a command is received from the client application to transmit a request to the server application,

determining the next sequence number [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

calling the wrap request function to wrap the request in a request message having a header containing the sequence number and a message type code indicating that the request message contains a request [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

calling the retransmit timer function to obtain a retransmit timer setting for setting a retransmit timer that counts down from the retransmit timer setting [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

starting the retransmit timer at the retransmit timer setting [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

sending the request message to the transport layer for transmission to the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], and

if the command to transmit the request asked that sequence number assigned to the message to be returned, then returning the sequence number to the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(d) when a message is received from the transport layer,

calling the parse function to obtain the message type and sequence number of the message, and if the message type and sequence number indicate that the message is a provisional reply to the request, then modifying the

retransmit timer to delay its expiration [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], and

if the message type and sequence number indicate that the message contains a reply to the request [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], returning the reply to the client application if:

the command to transmit the request specified that the reply be returned to the client application upon its receipt from the transport layer [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], or

the client application, since sending the command to transmit the request, has sent a command to return the reply upon its receipt from the transport layer [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], but otherwise storing the reply [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(e) when a command is received from the client application to return the reply to the request and the reply has been received and stored [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

returning the reply to the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(f) when the reply is returned to the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

calling the acknowledgement reply function to send a message to the transport layer for transmission to the server application acknowledging the

return of the reply to the client application, the message containing the sequence number and an indication that the message is an acknowledgement to the reply message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(g) if no message having a header containing the sequence number and a type code indicating that the message contains a reply has been received before the retransmit timer has expired [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], then repeatedly

calling the retransmit timer function to obtain a new retransmit timer setting for setting the retransmit timer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

setting and starting the retransmit timer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64], and

sending the request message to the transport layer for transmission to the server application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

until the retransmit timer has expired the maximum number of times or until a message having a header containing the sequence number and a type code indicating that the message contains a reply is received from the transport layer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

but if the retransmit timer has expired the maximum number of times and no such message has been received, then reporting a transmission error to the client application and destroying any subsequent messages having a header containing the sequence number until the sequence number is assigned to a new request message [Fig. 6g, 6h, item 670; Col. 11, lines 34-64];

(h) when a command is received from the client application to close the client socket [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

calling the close function to free up any resources allocated to support the client socket [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];
and in the server system:

(i) when a command is received from the server application to open a server socket,

receiving from the server application a set of pointers to a set of personality functions, which include

the open function [Fig. 4],

the parse function [Fig. 4],

a wrap reply function [Fig. 4],

a reply cache timer function [Fig. 4], and

the close function [Fig. 4], and

calling the open function to allocate resources needed to support the server socket [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(j) when a command is received from the server application to receive a request from the client application and then a message is received from the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

calling the parse function to obtain the message type and sequence number of the message and, if the message type indicates that the message

contains a request, the request, and then returning the request to the server application process [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(k) if, after delivery of the request to the server application, a command is received from the server application to send a provisional reply [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], then

 sending a message to the transport layer for transmission to the client application containing the sequence number and a message type code indicating that the message is a provisional reply to the request message [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20];

(l) when a command is received from the server application to transmit a reply to the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

 calling the wrap reply function to wrap the reply in a header containing the sequence number and a message type code indicating that the request message contains a reply [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20],

 calling the reply cache timer function to obtain a reply cache timer setting for setting a reply cache timer that counts down from the setting [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

 starting the reply cache timer [Fig. 6g, 6h, item 670; Col. 11, lines 34-64],

 sending the resulting message to the transport layer for transmission to the client application [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20], and

 caching the reply message [Fig. 6g, 6h, item 670; Col. 11, lines 34-64];

(m) when a further message is received from the client application, calling the parse function to obtain the message type and sequence number of the message and

if the message type indicates that the message contains a request and a cached reply message has the same sequence number, then resending the cached reply message to the transport layer for transmission to the client application [Fig. 6g, 6h, item 670; Col. 11, lines 34-64]], and

if a message contains an indication that the message is an acknowledgement to a cached reply message, then deleting everything except the sequence number from the cached reply message [Fig. 6g, 6h, item 670; Col. 11, lines 34-64]];

(n) when the reply cache timer expires, then deleting the cached reply message [Fig. 6g, 6h, item 670; Col. 11, lines 34-64]]; and

(o) when a command is received from the server application to close the server socket, calling the close function to free up any resources allocated by the personality functions [Figs. 5a-5d; Col. 9, line 24 - Col. 10, line 20].

Conclusion

Examiner's Note: Examiner has cited particular figures, columns, line numbers, and/or paragraphs in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety as potentially teaching all or part

of the claimed invention, as well as the context of the passage as taught by the prior art disclosed by the Examiner.

In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

Contacts

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tae K. Kim, whose telephone number is (571) 270-1979. The examiner can normally be reached on Monday - Friday (8:00 AM - 5:00 PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Thomas, can be reached on (571) 272-6776. The fax phone number for submitting all Official communications is (703) 872-9306. The fax phone number for submitting informal communications such as drafts, proposed amendments, etc., may be faxed directly to the examiner at (571) 270-2979.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at (866) 217-9197 (toll-free).

/Tae K. Kim/

Tae K. Kim
Examiner, Art Unit 2453

March 10, 2010

/Joseph Thomas/
Supervisory Patent Examiner, Art Unit 2453